

T-Mobile HPE Redfish Documentation

Contents

Module rf	3
Sub-modules	3
Module rf.change_password	4
Functions	4
Function change_password	4
Module rf.config	4
Module rf.create_user	4
Functions	4
Function create_user	4
Module rf.enable_sriov	5
Functions	5
Function enable_sriov	5
Module rf.get_bios	5
Functions	5
Function get_bios	5
Module rf.get_cpu	5
Functions	5
Function get_cpu	5
Module rf.get_cpu_blob	5
Functions	5
Function get_cpu_blob	5
Module rf.get_cpu_blob_alternative	5
Functions	5
Function get_cpu_blob	5
Module rf.get_disk_blob	6
Functions	6
Function get_disk_blob	6
Module rf.get_disk_capacity	6
Functions	6
Function get_disk_capacity	6
Module rf.get_disk_count	6
Functions	6
Function get_disk_count	6
Module rf.get_eth	6
Functions	6
Function get_eth	6

Module rf.get_firmware	7
Functions	7
Function get_firmware	7
Module rf.get_mem_blob	7
Functions	7
Function get_mem_blob	7
Module rf.get_mem_blob_alternative	7
Functions	7
Function get_mem_blob	7
Module rf.get_memory_total	7
Functions	7
Function get_memory_total	7
Module rf.get_model	8
Functions	8
Function get_model	8
Module rf.get_network_adapters_blob	8
Functions	8
Function get_network_adapters_blob	8
Module rf.get_nic_blob	8
Functions	8
Function get_nic_blob	8
Module rf.get_nic_names	8
Functions	8
Function get_nic_names	8
Module rf.get_status	9
Functions	9
Function get_status	9
Module rf.get_tag	9
Functions	9
Function get_tag	9
Module rf.get_uuid	9
Functions	9
Function get_uuid	9
Module rf.login	9
Functions	9
Function login	9
Module rf.reboot_server	9
Functions	9
Function reboot_server	9
Module rf.set_boot_order	10
Functions	10
Function set_boot_order	10
Module rf.set_dns	10
Functions	10
Function set_dns	10

Module <code>rf.set_firmware</code>	10
Functions	10
Function <code>set_firmware</code>	10
Module <code>rf.set_hostname</code>	10
Functions	10
Function <code>set_hostname</code>	10
Module <code>rf.set_jitter</code>	11
Functions	11
Function <code>set_jitter</code>	11
Module <code>rf.set_license_key</code>	11
Functions	11
Function <code>set_license_key</code>	11
Module <code>rf.set_ntp</code>	11
Functions	11
Function <code>set_ntp</code>	11
Module <code>rf.set_power_options</code>	11
Functions	11
Function <code>set_power_options</code>	11
Module <code>rf.set_snmp</code>	12
Functions	12
Function <code>set_snmp</code>	12
Module <code>rf.set_snmp_alerts</code>	12
Functions	12
Function <code>set_snmp_alerts</code>	12
Module <code>rf.set_timezone</code>	12
Functions	12
Function <code>set_timezone</code>	12
Module <code>rf.set_turbo</code>	12
Functions	12
Function <code>set_turbo</code>	12
Module <code>rf.set_virtualization</code>	13
Functions	13
Function <code>set_virtualization</code>	13

Module `rf`

Sub-modules

- [rf.change_password](#)
- [rf.config](#)
- [rf.create_user](#)
- [rf.enable_sriov](#)
- [rf.get_bios](#)
- [rf.get_cpu](#)
- [rf.get_cpu_blob](#)
- [rf.get_cpu_blob_alternative](#)
- [rf.get_disk_blob](#)
- [rf.get_disk_capacity](#)
- [rf.get_disk_count](#)

- `rf.get_eth`
- `rf.get_firmware`
- `rf.get_mem_blob`
- `rf.get_mem_blob_alternative`
- `rf.get_memory_total`
- `rf.get_model`
- `rf.get_network_adapters_blob`
- `rf.get_nic_blob`
- `rf.get_nic_names`
- `rf.get_status`
- `rf.get_tag`
- `rf.get_uuid`
- `rf.login`
- `rf.reboot_server`
- `rf.set_boot_order`
- `rf.set_dns`
- `rf.set_firmware`
- `rf.set_hostname`
- `rf.set_jitter`
- `rf.set_license_key`
- `rf.set_ntp`
- `rf.set_power_options`
- `rf.set_snmp`
- `rf.set_snmp_alerts`
- `rf.set_timezone`
- `rf.set_turbo`
- `rf.set_virtualization`

Module `rf.change_password`

Functions

Function `change_password`

```
def change_password(rfo, username, password, api=1)
```

Change iLO user account password

Parameters: `rfo` (object): Redfish login session `username` (str): User name `password` (str): New password

Returns: str: iLO response status

Module `rf.config`

Module `rf.create_user`

Functions

Function `create_user`

```
def create_user(rfo, username, password, role, api=1)
```

Create iLO user account

Parameters: `rfo` (object): Redfish login session `username` (str): User name `password` (str): Password `role` (str): Administrator, ReadOnly or Operator

Returns: str: iLO response status

Module `rf.enable_sriov`

Functions

Function `enable_sriov`

```
def enable_sriov(rfo, enable='Enabled', api=1, unit=1)
```

Enable SRIOV

Parameters: rfo (object): Redfish login session enable (str): Enabled, Disabled api (int): API Value unit (int): Unit Value

Returns: str: iLO response status

Module `rf.get_bios`

Functions

Function `get_bios`

```
def get_bios(rfo, api=1, unit=1)
```

This function fetches the BIOS VERSION of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ Information is in the following JSON snippet. "BiosVersion": ""

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: string: Bios Version

Module `rf.get_cpu`

Functions

Function `get_cpu`

```
def get_cpu(rfo, api=1, unit=1)
```

This function fetches the CPU model of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ Information is in the following JSON snippet. "ProcessorSummary": { "Model" }

Parameters: object: Redfish Client Login Object int: API Value int: Unit Value

Returns: string: CPU Model

Module `rf.get_cpu_blob`

Functions

Function `get_cpu_blob`

```
def get_cpu_blob(rfo, api=1, unit=1)
```

Aggregate CPU information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module `rf.get_cpu_blob_alternative`

Functions

Function `get_cpu_blob`

```
def get_cpu_blob(rfo, api=1, unit=1)
```

Display aggregate CPU information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module **rf.get_disk_blob**

Functions

Function **get_disk_blob**

```
def get_disk_blob(rfo, api=1, unit=1)
```

Aggregate disk information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module **rf.get_disk_capacity**

Functions

Function **get_disk_capacity**

```
def get_disk_capacity(rfo, api=1, unit=1)
```

This function fetches the Disk Capacity of the server. URL is https://IP_ADDRESS/redfish/v1/chassis/1/ Information is in the following JSON snippet. “Links”: { “Drives”:{}

NOTE: There could be multiple drives in the server.

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: int: Combined Disk Capacity (Bytes)

Module **rf.get_disk_count**

Functions

Function **get_disk_count**

```
def get_disk_count(rfo, api=1, unit=1)
```

This function fetches the number of disks in the server. URL is https://IP_ADDRESS/redfish/v1/chassis/1/ Information is in the following JSON snippet. “Links”: { “Drives”:{}

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: int: disk count

Module **rf.get_eth**

Functions

Function **get_eth**

```
def get_eth(rfo, api=1, unit=1, nic_name='')
```

This function fetches the information of a specific NIC on the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ MODEL information is at the top level of the JSON Hierarchy

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: names of the NICS on the server

Module `rf.get_firmware`

Functions

Function `get_firmware`

```
def get_firmware(rfo, api=1, unit=1)
```

This function fetches the FIRMWARE of the server. URL is https://IP_ADDRESS/redfish/v1/chassis/1/Information is in the following JSON snippet. OEM: { Hpe: { Firmware: { "PlatformDefinitionTable": { "Current": { "VersionString" }}}}}

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: string: Firmware

Module `rf.get_mem_blob`

Functions

Function `get_mem_blob`

```
def get_mem_blob(rfo, api=1, unit=1)
```

Aggregate memory DIMM information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module `rf.get_mem_blob_alternative`

Functions

Function `get_mem_blob`

```
def get_mem_blob(rfo, api=1, unit=1)
```

Display aggregate memory information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module `rf.get_memory_total`

Functions

Function `get_memory_total`

```
def get_memory_total(rfo, api=1, unit=1)
```

This function fetches the TOTAL MEMORY installed in the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/Information is in the following JSON snippet. "MemorySummary": { "TotalSystemMemoryGiB": }

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: int: Total Memory

Module `rf.get_model`

Functions

Function `get_model`

```
def get_model(rfo, api=1, unit=1)
```

This function fetches the MODEL of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ MODEL information is at the top level of the JSON Hierarchy

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: str: Model Info

Module `rf.get_network_adapters_blob`

Functions

Function `get_network_adapters_blob`

```
def get_network_adapters_blob(rfo, api=1, unit=1)
```

Aggregate network adapter information

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: JSON

Module `rf.get_nic_blob`

Functions

Function `get_nic_blob`

```
def get_nic_blob(rfo, api=1, unit=1)
```

This function fetches information about EACH of the network interfaces of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/EthernetInterfaces/

NOTE: There could be multiple interfaces in the server.

Parameters: object: Redfish Client Login Object int: API Value int: Unit Value

Returns: list: JSON - Details of each of the Ethernet Interfaces in the Server

Module `rf.get_nic_names`

Functions

Function `get_nic_names`

```
def get_nic_names(rfo, api=1, unit=1)
```

This function fetches the names of the NICS on the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ MODEL information is at the top level of the JSON Hierarchy

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: list: Names of the NICS on the server

Module `rf.get_status`

Functions

Function `get_status`

```
def get_status(rfo, api=1, unit=1)
```

This function fetches the Health Status of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ Information is in the following JSON snippet. "Status": { "Health": }

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: str: Health Status

Module `rf.get_tag`

Functions

Function `get_tag`

```
def get_tag(rfo, api=1, unit=1)
```

This function fetches the ASSET TAG of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ Information is in the following JSON snippet. "AssetTag": ""

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: str: Asset Tag

Module `rf.get_uuid`

Functions

Function `get_uuid`

```
def get_uuid(rfo, api=1, unit=1)
```

This function fetches the UUID of the server. URL is https://IP_ADDRESS/redfish/v1/systems/1/ Information is in the following JSON snippet. "UUID":

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: str: UUID

Module `rf.login`

Functions

Function `login`

```
def login(un, pw, url)
```

Login to Redfish API un: (str) User name pw: (str) User password url: (str) HTTPS RedFish API URL

Return: Redfish object

Module `rf.reboot_server`

Functions

Function `reboot_server`

```
def reboot_server(rfo, api=1, unit=1)
```

Reboot server

Parameters: rfo (object): Redfish Client Login Object api (int): API Value unit (int): Unit Value

Returns: str: Reboot status

Module **rf.set_boot_order**

Functions

Function **set_boot_order**

```
def set_boot_order(rfo, order, api=1, unit=1)
```

Set processor turbo boost technology

Parameters: rfo (object): Redfish login session order (list): Ordered list of devices api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_dns**

Functions

Function **set_dns**

```
def set_dns(rfo, dns, api=1, unit=1)
```

Set Primary DNS Servers

Parameters: rfo (object): Redfish session dns (array): List of DNS server IP addresses api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_firmware**

Functions

Function **set_firmware**

```
def set_firmware(rfo, firmware, update_repo=True, update_target=False)
```

Upload firmware to iLO repository and optionally apply

Parameters: rfo (object): Redfish session firmware (str): Path to file location (ie “data/ilo5_215.bin”) update_repo (boolean): Upload firmware to iLO repository update_target (boolean): Apply firmware

Returns: str: iLO response status

Module **rf.set_hostname**

Functions

Function **set_hostname**

```
def set_hostname(rfo, hostname, api=1, unit=1)
```

Set iLO hostname

Parameters: rfo (object): Redfish login session hostname (str): Host name api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_jitter**

Functions

Function **set_jitter**

```
def set_jitter(rfo, mode, api=1, unit=1)
```

Set processor jitter control

Parameters: rfo (object): Redfish login session mode (str): Disabled, Auto-tuned, Manual-tuned api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_license_key**

Functions

Function **set_license_key**

```
def set_license_key(rfo, key, api=1, unit=1)
```

Set iLO license key

Parameters: rfo (object): Redfish session key (str): License key api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_ntp**

Functions

Function **set_ntp**

```
def set_ntp(rfo, ntp, api=1, unit=1)
```

Set static NTP servers

Parameters: rfo (object): Redfish login session ntp (array): Array of IP addresses api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module **rf.set_power_options**

Functions

Function **set_power_options**

```
def set_power_options(rfo, mode, api=1, unit=1)
```

Set power mode options

Parameters: rfo (object): Redfish login session mode (str): Power mode TODO: power modes available api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module `rf.set_snmp`

Functions

Function `set_snmp`

```
def set_snmp(rfo, destinations, api=1, unit=1)
```

Set SNMP Alert Destinations

Parameters: rfo (object): Redfish login session destinations (array): The IP address or FQDN of remote management system that receive SNMP alerts. api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module `rf.set_snmp_alerts`

Functions

Function `set_snmp_alerts`

```
def set_snmp_alerts(rfo, alert, communities, api=1, unit=1)
```

Set SNMP enable alerts and read communities

Parameters: rfo (object): Redfish login session alert (boolean): Enable alerts community (array): SNMP read communities api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module `rf.set_timezone`

Functions

Function `set_timezone`

```
def set_timezone(rfo, tz, api=1, unit=1)
```

Set iLO timezone

Parameters: rfo (object): Redfish login session tz (str): Examples: UtcM10:UTC-10:00, Hawaii UtcM8:UTC-08:00, Pacific Time(US & Canada) UtcM7:UTC-07:00, Mountain Time (US & Canada) UtcM6:UTC-06:00, Central America, Central Time(US & Canada) UtcM5:UTC-05:00, Eastern Time(US & Canada) api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module `rf.set_turbo`

Functions

Function `set_turbo`

```
def set_turbo(rfo, mode, api=1, unit=1)
```

Set processor turbo boost technology

Parameters: rfo (object): Redfish login session mode (str): Enabled, Disabled api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Module `rf.set_virtualization`

Functions

Function `set_virtualization`

```
def set_virtualization(rfo, mode, api=1, unit=1)
```

Enable processor hypervisor virtualization

Parameters: rfo (object): Redfish login session mode (str): Enabled, Disabled api (int): API version unit (int): Enumerated component unit

Returns: str: iLO response status

Generated by *pdoc* 0.8.1 (<https://pdoc3.github.io>).